Processing large scale streaming data from high energy physics workflows

Darren Kerbyson, Mahantesh Halappanavar, Malachi Schram, Kevin Barker,
Nathan R. Tallent, Jian Yin, Eric Stephan, Ryan Friese, Luis de la Torre
High Performance Computing, Pacific Northwest National Laboratory, Richland, USA

The Belle II experiments stream massive amounts of data. Motivated by Belle II's complex data interactions, we are developing techniques for improving the efficiency of stream-based workflows. Designed to probe the interactions of the fundamental constituents of our universe, the Belle II experiments will generate 25 petabytes of raw data per year. During the course of the experiments, the necessary storage is expected to reach over 350 petabytes.[1] Data is generated by the Belle II detector, Monte Carlo simulations, and user analysis. The detector's experimental data is processed and re-processed through a complex set of operations, which are followed by analysis in a collaborative manner. Users, data, storage and compute resources are geographically distributed across the world creating a complex data intensive workflow.

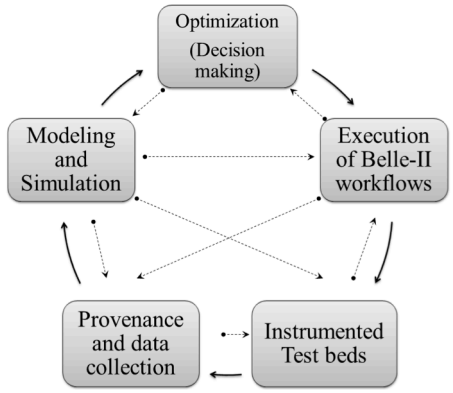

Figure 1. Illustration of an integrated approach for efficient execution of Belle-II workflows.

The Belle II computing model supports the following tasks: (*i*) RAW data processing; (*ii*) Monte Carlo production; (*iii*) physics analysis; and (*iv*) data storage and data archiving. These tasks consume four sources of streaming and non-streaming data: (*a*) Raw data captured from the Belle II detector by the data acquisition; and (*b*) processed/refined raw data used by physicist; (*c*) data generated from the Monte Carlo simulations; and (*d*) user analysis data. The first two data sources involve streaming. The latter two involve such large data volumes that achieving effective response times requires streaming data rates.

Each type of data is stored, accessed and processed in a geographically distributed manner. The coordination of the first three types of data is managed centrally; however, user analyses are chaotic in nature. The computational and data needs for Belle II experiments are satisfied through a geographically distributed network of storage and compute clusters, in addition to on-demand use of cloud computing platforms. In the first 3 years of data taking, KEK Data Center in Japan and the PNNL Data Center in Washington, USA will host the raw data on online disk and offline tape storage. Storage for processed RAW data, Monte Carlo samples and user analysis samples will be provided by regional data centers located in Asia, North America, and Europe. Computational resources (dedicated and non-dedicated compute clusters) are available in Asia, North America, and Europe. Additionally, commercial clouds such as Amazon EC2, and non-commercial clouds such as Cracow are also available for Belle II experiments. These resources are interconnected through high-speed networks such as the SINET, ESnet, GEANT, among others [2].

Motivated by the complex workflows within Belle II, we propose an approach for efficient execution of workflows on distributed resources that integrates provenance, performance modeling, and optimization-based scheduling. Figure 1 illustrates our approach. The key components of this framework include modeling and simulation methods to quantitatively predict workflow component behavior; optimized decision making such as choosing an optimal subset of resources to meet demand, assignment of tasks to resources, and placement of data to minimize data movement; prototypical testbeds for workflow execution on distributed resources; and provenance methods for collecting appropriate performance data.

---

[1] D. M. Asner, E. Dart, and T. Hara. Belle II experiment network and computing. Technical Report arXiv:1308.0672. PNNL-SA-97204, Aug 2013. Contributed to CSS2013 (Snowmass).

[2] T. Hara. Belle II computing and network requirements, Proc. of the Asia-Pacific Advanced Network, 2014. 115-122.

Belle II workflows necessitate decision making at several levels. We therefore present a hierarchical framework for data driven decision making, illustrated in Figure 2. Given an estimated demand for compute and storage resources for a period of time, the first (top) level of decision making involves identifying an optimal (sub)set of resources that can meet the predicted demand. We use the analogy of unit commitment problem in electric power grids to solve this problem[3]. Once a cost-efficient set of resources are chosen, the next step is to assign individual tasks from the workflow to specific resources. For Belle II, we consider the situation of Monte Carlo campaigns that involves a set of independent tasks (bag-of-tasks) that need to be assigned on distributed resources. We use a semi-matching based approach for efficient computing of approximate solutions. These two decisions are made at the system level. Further, there are several decisions that need to be made at a local level such as, optimal scheduling of tasks on a given resource optimal placement of data, and scheduling of data movement to minimize file system and network congestion.

In order to support accurate and efficient scheduling, predictive performance modeling is employed to rapidly quantify expected task performance across available hardware platforms. The goals of this performance modeling work are to gain insight into the relationship between workload parameters, system characteristics, and performance metrics of interest (e.g., task throughput or scheduling latency); to characterize observed performance; and to guide future and runtime optimizations (including task/module scheduling). Of particular interest, these quantitative and predictive models provide the cost estimates to the higher-level task scheduling algorithms, allowing the scheduler to make informed decisions concerning the optimal resources to utilize for task execution.

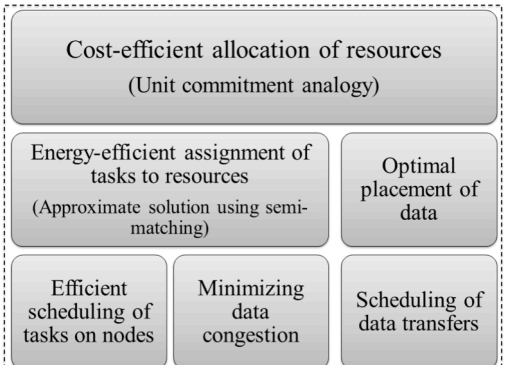

Figure 2. An overview of the hierarchical scheduling framework.

Modeling the performance of a complete workflow requires modeling the constituent tasks comprising that workflow. We follow a white-box approach, in that each task is analyzed to discover the activities that make up its execution, including computation, communication (for parallel tasks), memory access, and file I/O. To assist in this effort, we are employing the PALM tool[4]. PALM relies on program annotations to identify and make connections between observed behavior, code structure, and input characteristics. The resulting models are static representations of each tasks dynamic behavior, and are parameterized not only in terms of input characteristics, but also in terms of each execution platforms performance capabilities. The output of each model is a prediction of behavior on each available platform within the larger distributed environment.

Critical to the success of our modeling methodology is rigorous model validation. To date, we have utilized two PNNL resources as test bed platforms, allowing us to compare predicted and empirical performance. The SeaPearl cluster consists of 32 Intel Ivy Bridge nodes, each containing two 10-core sockets running at 2.1 GHz and containing 128 GB of DDR3 memory. Nodes are connected using 4xQDR InfiniBand. The SeaPearl cluster is equipped with Penguin Computing PowerInsight 2.1, providing up to 21 channels of power and thermal data per node sampled at 1 KHz per channel. Characterizing task execution using data derived from the PowerInsight sensors, it is possible to develop models that capture power and energy information and use these metrics in task scheduling.

[3] M Halappanavar, M Schram, L de la Torre, K Barker, N Tallent, and D Kerbyson. "Towards Efficient Scheduling of Data Intensive High Energy Physics Workflows." In WORKS 2015 Workshop, held in conjunction with the International Conference for High Performance Computing, Networking, Storage and Analysis (SC15).

[4] N. Tallent and A. Hoisie. Palm: Easing the burden of analytical performance modeling. In Proc. of the 28th ACM International Conference on Supercomputing, pages 221-230, 2014.